

Model Development Phase Template

Date	July 2024
Team ID	739708
Project Title	Ecoforecast AI-powered prediction of carbon monoxide levels
Maximum Marks	10 Marks

Initial Model Training Code, Model Validation and Evaluation Report

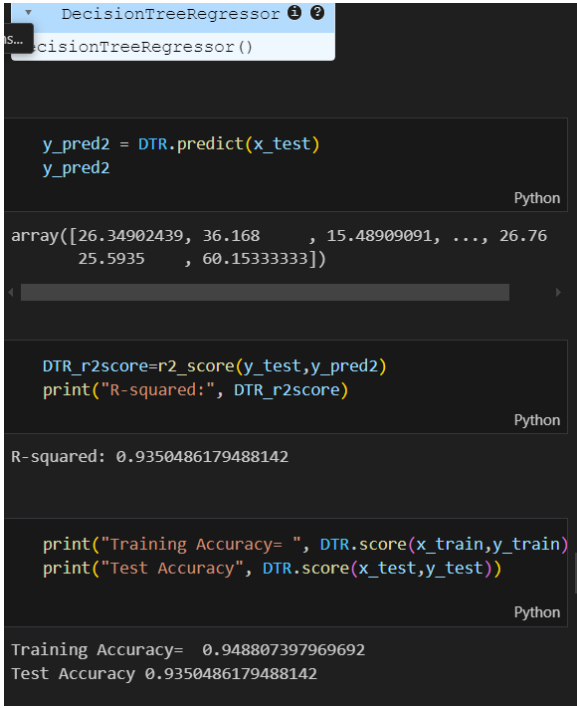
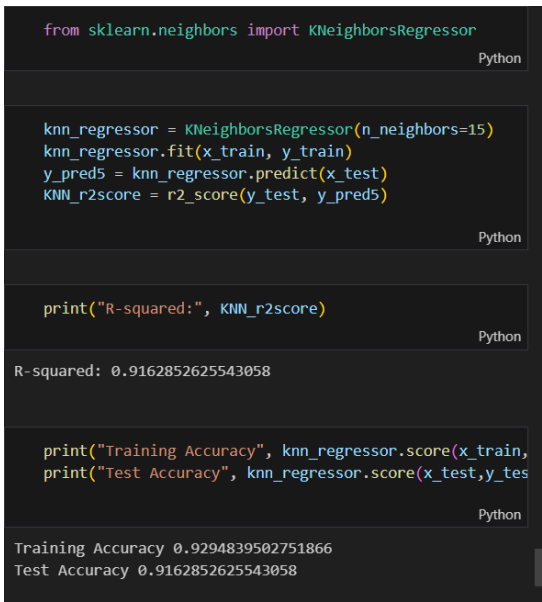
The initial model training code will be showcased in the future through a screenshot. The model validation and evaluation report will include a summary and training and validation performance metrics for multiple models, presented through respective screenshots.

Initial Model Training Code (5 marks):

Paste the screenshot of the model training code

Model Validation and Evaluation Report (5 marks):

Model	Summary	Training and Validation Performance Metrics
Model 1	Logistic regression model typically include accuracy, precision, recall, r2_score to evaluate its predictive performance and generalization capability.	<pre>from sklearn.linear_model import LinearRegression LR = LinearRegression() LR.fit(x_train, y_train)</pre> <p>Python</p> <p>LinearRegression</p> <pre>y_pred = LR.predict(x_test) y_pred</pre> <p>Python</p> <pre>array([43.75486282, 42.38330191, 28.99622554, ..., 49.5244, 21.4783531 , 42.3413951])</pre> <pre>print("Training Accuracy=", LR.score(x_train, y_train)) print("Test Accuracy ", LR.score(x_test,y_test))</pre> <p>Python</p> <p>Training Accuracy= 0.2256505204635355 Test Accuracy 0.22101875220973688</p>
Model 2	Random forest classifier model often encompass accuracy, precision, recall, r2_score to measure its prediction quality and robustness.	<pre>klearn.ensemble import RandomForestRegressor RandomForestRegressor(n_estimators = 20, random_state = t(x_train, y_train) 1 = RFR.predict(x_test) score = r2_score(y_test,y_pred1)</pre> <p>Python</p> <pre>from sklearn import metrics print('R_squared: ', RFR_r2score)</pre> <p>Python</p> <p>R_squared: 0.935374935760041</p> <pre>print("Training Accuracy", RFR.score(x_train,y_train)) print("Test Accuracy", RFR.score(x_test,y_test))</pre> <p>Python</p> <p>Training Accuracy 0.9479937857412938 Test Accuracy 0.935374935760041</p>

<p>Model 3</p>	<p>Decision tree classifier model commonly include accuracy, precision, recall, r2_score which help assess the model's prediction accuracy and generalizability.</p>	 <pre> DecisionTreeRegressor DecisionTreeRegressor() y_pred2 = DTR.predict(x_test) y_pred2 array([[26.34902439, 36.168 , 15.48909091, ..., 26.76 25.5935 , 60.15333333]]) DTR_r2score=r2_score(y_test,y_pred2) print("R-squared:", DTR_r2score) R-squared: 0.9350486179488142 print("Training Accuracy= ", DTR.score(x_train,y_train)) print("Test Accuracy", DTR.score(x_test,y_test)) Training Accuracy= 0.948807397969692 Test Accuracy 0.9350486179488142 </pre>
<p>Model 4</p>	<p>K-nearest neighbors classifier model typically include accuracy, precision, recall, r2_score to evaluate its prediction performance and generalization ability.</p>	 <pre> from sklearn.neighbors import KNeighborsRegressor knn_regressor = KNeighborsRegressor(n_neighbors=15) knn_regressor.fit(x_train, y_train) y_pred5 = knn_regressor.predict(x_test) KNN_r2score = r2_score(y_test, y_pred5) print("R-squared:", KNN_r2score) R-squared: 0.9162852625543058 print("Training Accuracy", knn_regressor.score(x_train, print("Test Accuracy", knn_regressor.score(x_test,y_test) Training Accuracy 0.9294839502751866 Test Accuracy 0.9162852625543058 </pre>